

NO EXCUSES

CONCEPT TO CASH EVERY WEEK

11-Mar-2009



energizedwork.com



Gus Power



Kris Lander



THE ENERGIZED WAY

MAXIMIZING CLIENT PROFIT



THROUGHPUT

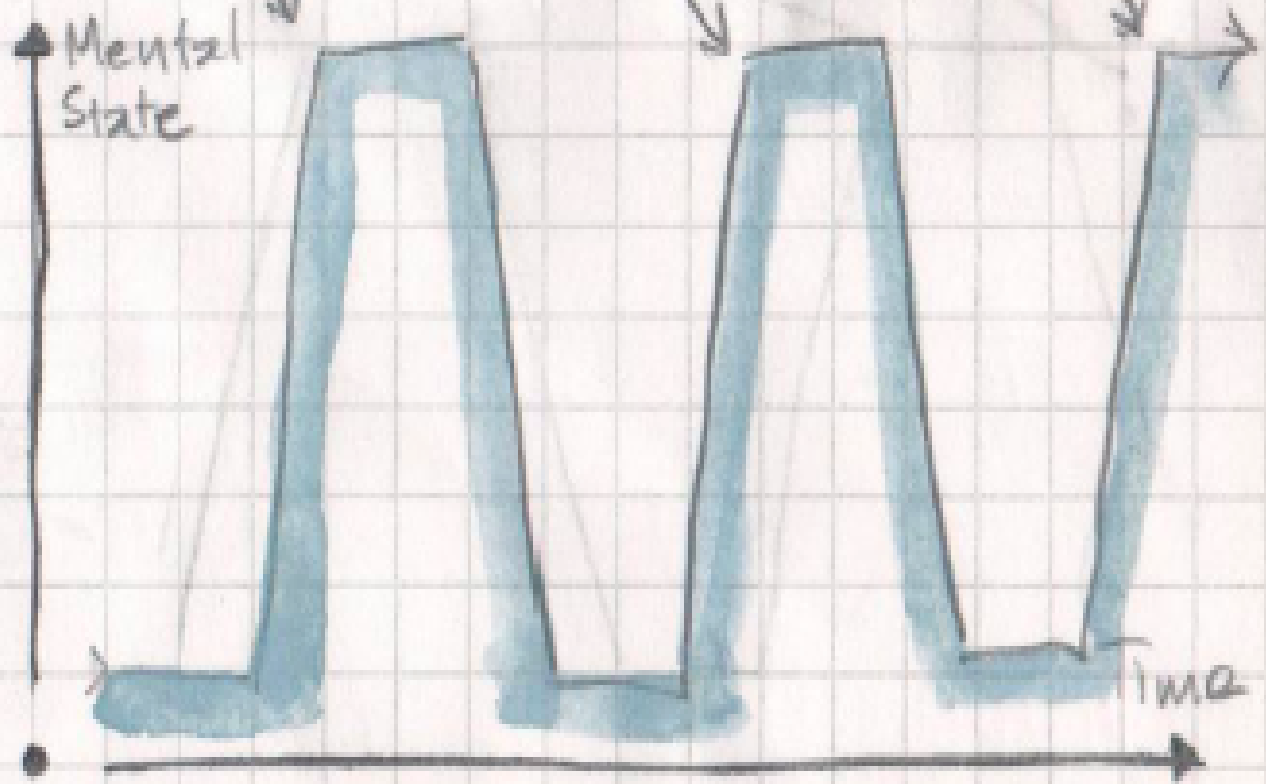
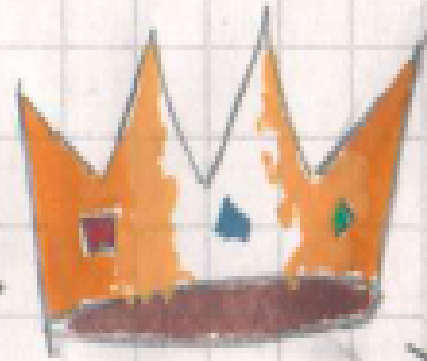
TESTABILITY

REPEATABILITY

MANEUVERABILITY



Timebox and Small Releases and Protect Iteration



UK's premier Entertainment site

180M page impressions / month

3.7M unique users / month

Average page weight of 2MB

9 wks:
Standalone
alpha

17 wks:
Parallel-
flight beta

20 wks:
Live replacing
all existing sites

69 wks:
Still adding
new features



**KEEP
IT
MOVING**



Sustainable throughput

Follow the money and 'ship' every iteration to production

Financials

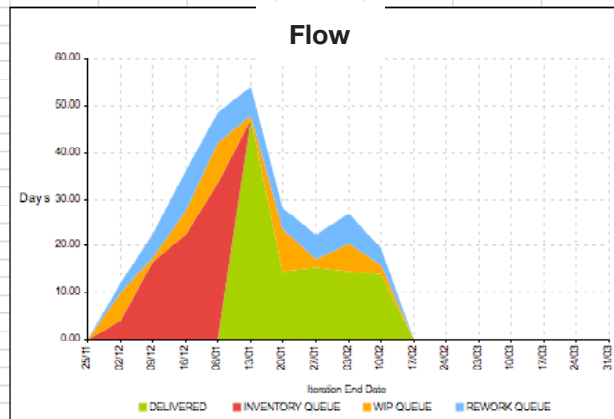
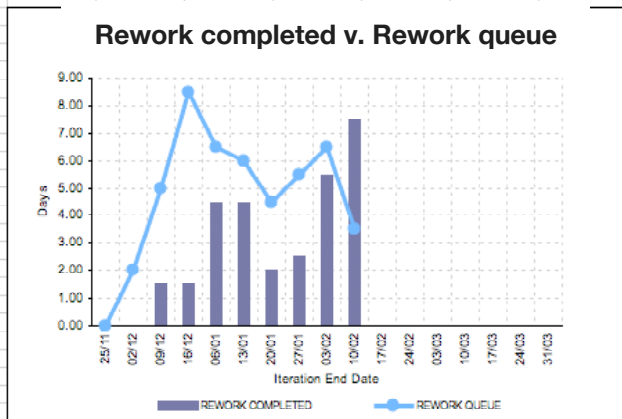
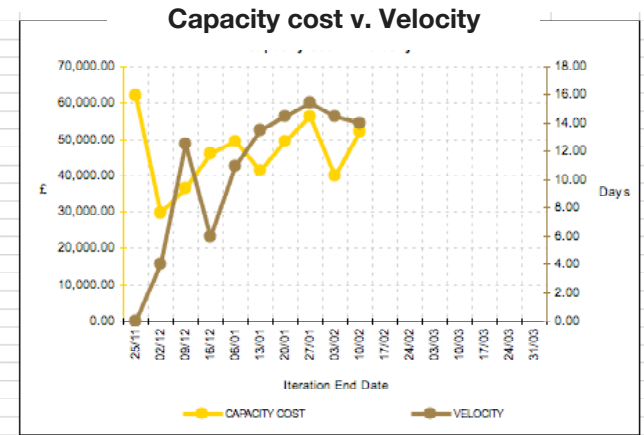
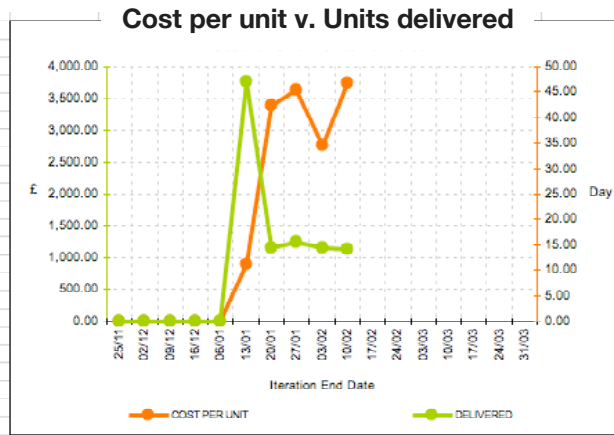
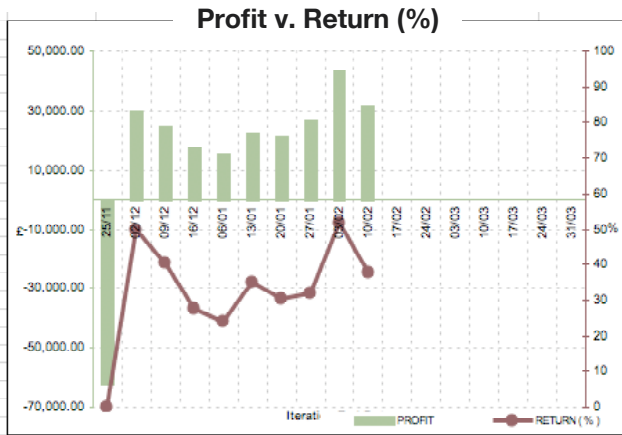
Minimize work in process

Avoid inventory

Keep it visible with a £ value



KEEP IT MOVING



1-Week iteration

Focus relentlessly on delivering profitable features every week

Surfaces waste

Flushes out breaks in value stream

Start thinking about 'done' straight away

Regulates pressure over longer periods

Start on Wednesdays



Automated deployment

Maintain a 1-click deploy that requires no manual intervention

Test, Perf, Demo, CI (scaled versions of Prod)

Robust and repeatable

A / B legs; no down time

Hudson, Gant

Use separate queue

Automated schema/data migration

Check deployed version



- [Back to Project](#)
- [Status](#)
- [Changes](#)
- [Console Output](#)**
- [Tag this build](#)
- [Previous Build](#)

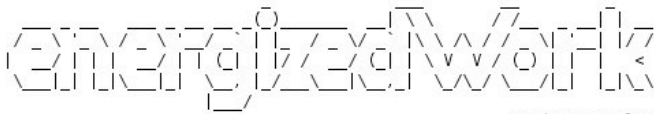
Console Output

```
started
Updating https://svn.energizedwork.com/energizedwork/product/commuting-site/commutineer
At revision 5855
+ grails -Dgrails.env=beta deploy
```

```
Welcome to Grails 1.0.3 - http://grails.org/
Licensed under Apache Standard License 2.0
Grails home is set to: /opt/grails
```

```
Base Directory: /opt/hudson/.hudson/jobs/beta/workspace/commutineer
Running script /opt/hudson/.hudson/jobs/beta/workspace/commutineer/scripts/Deploy.groovy
Environment set to beta
Found application events script
Found events script in plugin code-coverage-0.8.3
  Reading application configuration from /opt/hudson/.hudson/jobs/beta/workspace/commutineer/grails-app/conf/Config.
  Checking for system property overrides (i.e. -Dserver.property=value) ...
  Verifying deployment configuration ...
  Settings: ...
  webservers=beta.commutineer.com
  deploymentkey=/opt/hudson/.hudson/jobs/beta/workspace/commutineer/environment/beta.deploy
  Updating product on target webservers ...
  Updating product on beta.commutineer.com ...
```

```
[sshexec] Connecting to beta.commutineer.com:22
Welcome to the
```



product overlay

```
Unauthorized Access Prohibited.
Have a nice day.
```

```
receiving file list ...
7 files to consider
net-www/commutineer/
net-www/commutineer/Manifest
   0  0%  0.00kB/s   0:00:00
 1218 100%  1.16MB/s   0:00:00 (xfer#1, to-check=3/7)
0 files...
```

All	Backups	Code	Demo	Deploy	+
S		W		Job ↓	
				demo	
				publish	
				qa	
				selenium-firefox-deploy	
				selenium-ie7-deploy	

Icon: [S](#) [M](#) [L](#)



KEEP IT MOVING

Iterate to get something out there

Enrich the functional depth of features iteratively

Business learns what it needs

‘Cheapest’ solution first



Dealing with bugs

Fix bugs as soon as they're discovered

No bug backlog and no tracking tool

Onsite customer says what's really a bug

Within slice, interrupt pair and fix in story

Outside story, write pink card with visible priority at top of board



Managing technical debt

Don't get into debt. If you do, repay some of it every iteration

Team decision

Confront value fetishists

Blue card always visible with £ value





**KEEP
IT
WORKING**



Test-driven development

Working from the outside in, drive with tests to keep code habitable, testable and maneuverable

Low cost of change

Maintain options with 'succession'

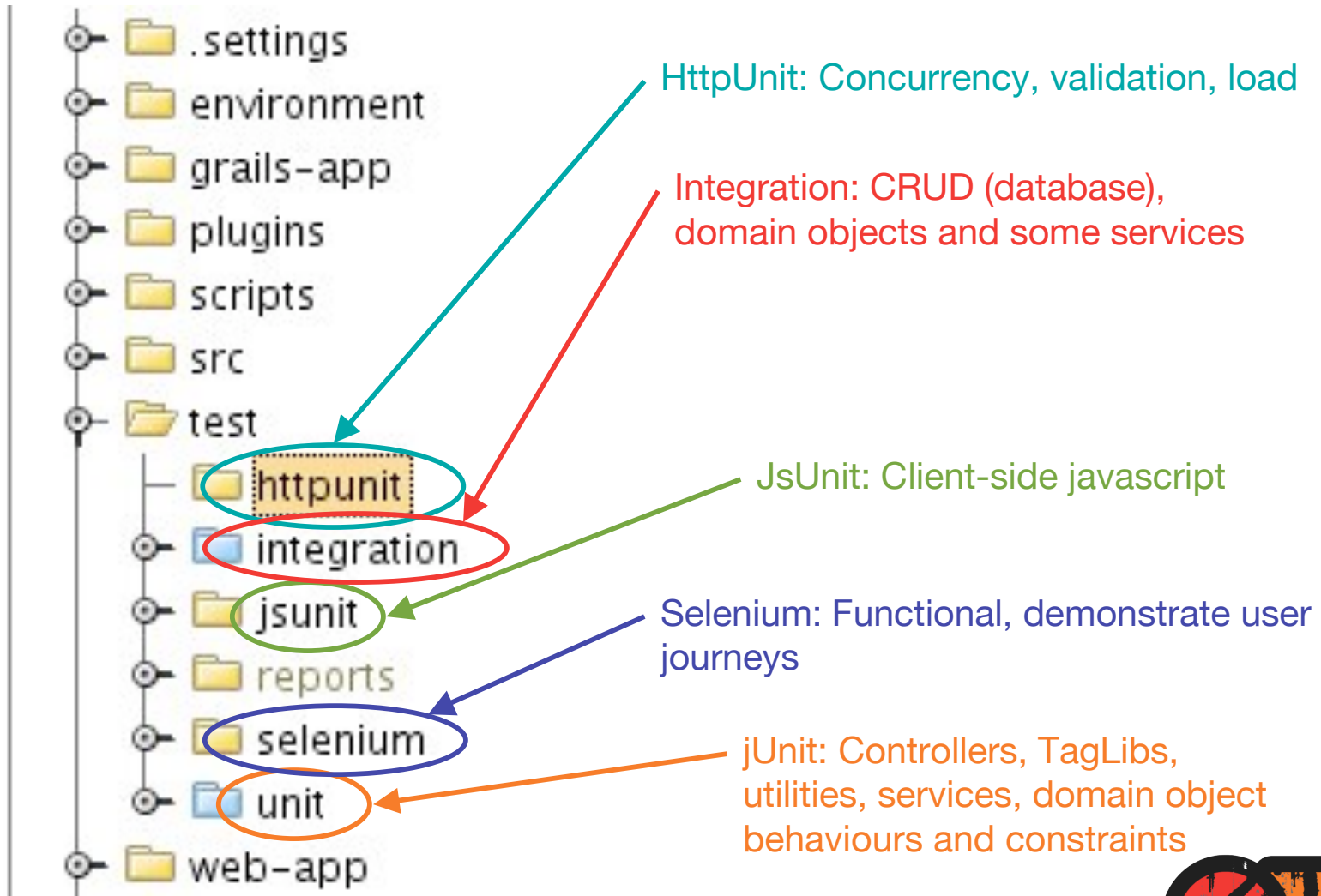
Do the valuable refactorings

Acceptance criteria create common language

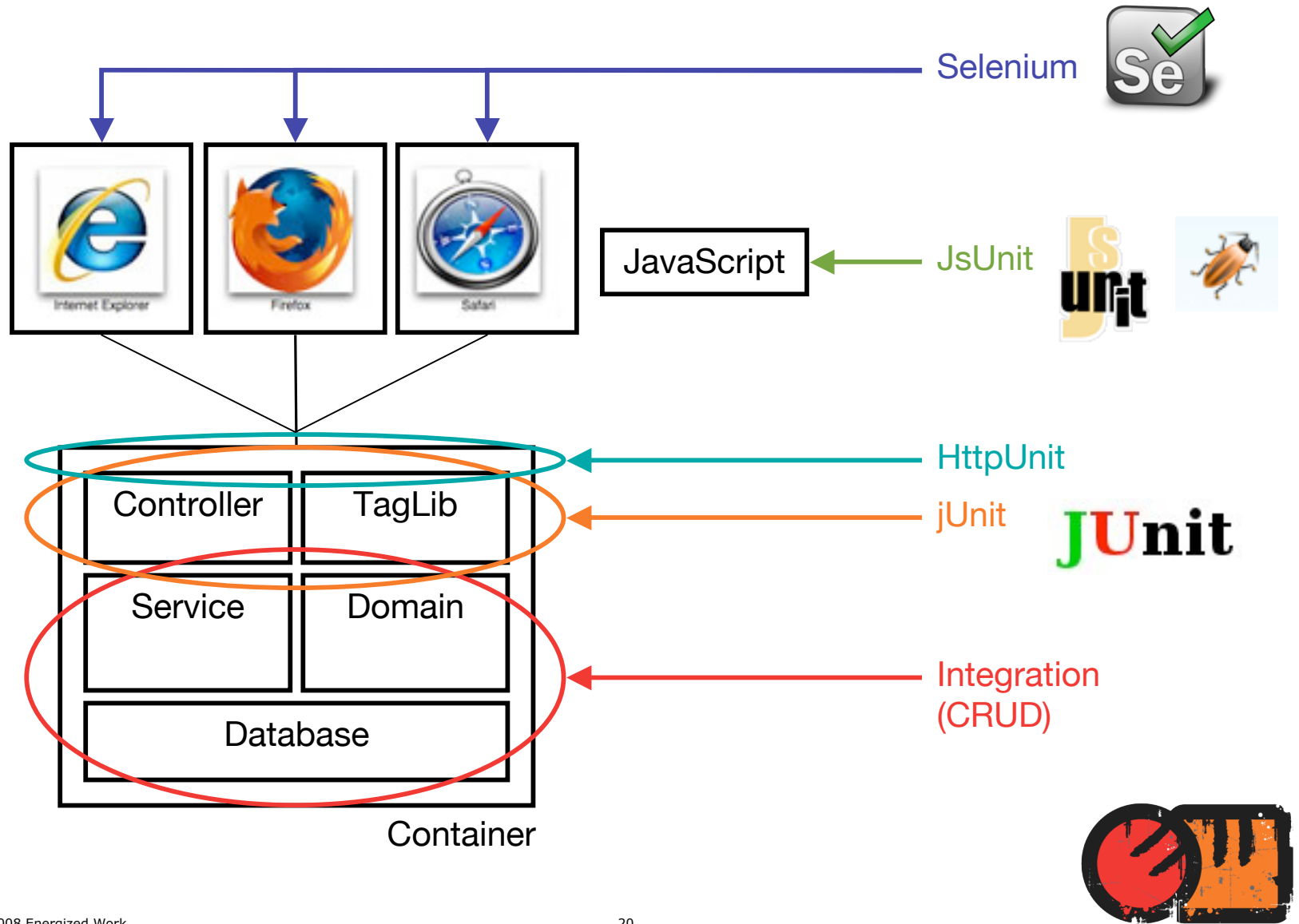
Executable acceptance tests



KEEP IT WORKING



KEEP IT WORKING



KEEP IT WORKING



Revision 5966

[/skunkworks/grails/plugins](#)

[\[Parent Directory\]](#)

[.plugin-meta/](#)

[ebuild/](#)

[grails-ew-selenium/](#)

[httpunit/](#)

[imageCut/](#)

[jamon/](#)

[jsunit/](#)

[lucidImages/](#)

[selenium/](#)

[serverInfo/](#)



KEEP IT WORKING

```
45
46 void testValidationSucceedsGivenValidAccountAndOpenPosition() {
47     Position position = new CustomerPosition(id: '123', open: true, quantity: 100)
48
49     mockDomain(Position, [position])
50
51     CloseTradeValidator validator = new CloseTradeValidator()
52     CloseCommand cmd = new CloseCommand(positionId: '123', quantityStr: '50')
53
54     validator.run(cmd)
55     assertFalse validator.hasErrors()
56 }
57
```

```
76     Position position = PositionBuilder.build(instrument: blotter.instrument, quantity: 1000)
77
78     subBlotter.calc = [
79         houseIsShort: { true }
80     ] as BlotterCalculator
```

```
99
100 void testClosePositionReturnsACloseCommand() {
101
102     CloseCommand cmd = new CloseCommand()
103
104     def validatorMock = new MockFor(CloseTradeValidator)
105     validatorMock.demand.run { }
106     validatorMock.demand.hasErrors() { return true }
107
108     validatorMock.use {
109         assertEquals cmd, positionService.close(cmd)
110     }
111 }
112
```



```
html head script
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5 <link rel="stylesheet" type="text/css" href="../../css/jsUnitStyle.css">
6 <script type="text/javascript" src="../../app/jsUnitCore.js"></script>
7 <script type="text/javascript" src="../../../../web-app/js/jquery-1.2.6.js"></script>
8 <script type="text/javascript" src="../../../../web-app/js/application.js"></script>
9 <script type="text/javascript">
10     function testThatFxRatesAreUpdated() {
11         var responseData = [{id: '12345', 'buy': '2.51', 'sell': '2.49'}]
12         fxlist.update(responseData);
13         assertEquals("2.51", $('#tr#12345 td.buy')[0].innerHTML);
14         assertEquals("2.49", $('#tr#12345 td.sell')[0].innerHTML);
15     }
16
17     function testThatFunctionCanHandleFxRatesThatDontExistOnPage() {
18         var responseData = [{id: '23456', 'buy': '2.51', 'sell': '2.49'}]
19         fxlist.update(responseData);
20     }
21 </script>
22 </head>
23 <body>
24 <table id=' fxRate'>
25 <tbody>
26 <tr id="12345">
27 <td class="sell"></td>
28 <td class="buy"></td>
29 </tr>
30 </tbody>
31 </table>
32 </body>
33 </html>
```

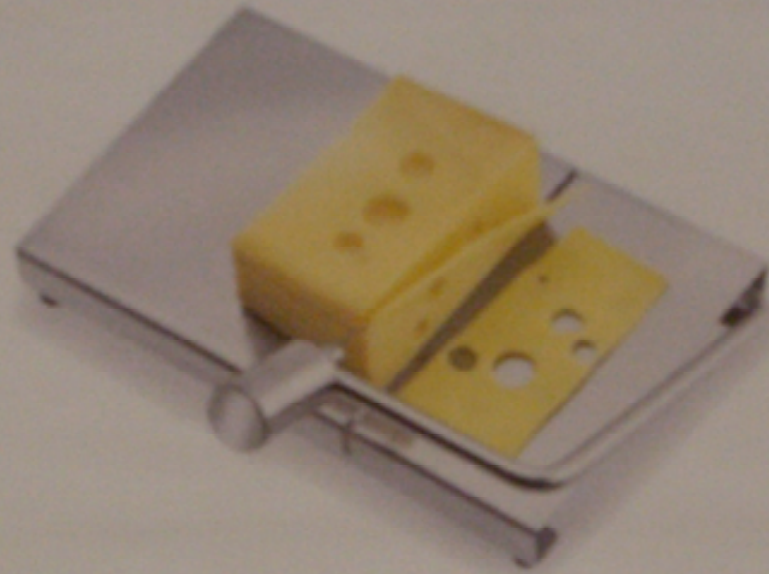


Continuous feedback

Develop stories in vertical slices to create opportunities for feedback ●

- Tester does exploratory testing
- Onsite customer steers feature evolution
- Interaction designer sees feature evolve
- Follow the dots to 'done'





THINK CHEESE, AND SLICE



DEALER MANUALLY BOOKS
HEDGE TRADES
(HEDGE TRADES)

SC
GP
KR
PB
GP
KR
MA
GP
RL



CREATE BROKER
ACCOUNTS.
(HEDGE TRADES)

PB
DN
SC
KR

2

TOD	TH	FR	M
2	1-5	1	

2

Pair-programming

Work in pairs all the time using 'Pomodoro' 25-minute time-boxing

Maintain energy levels and pace

Maintain focus on the acceptance criterion and getting the slice done





Collective ownership

Swap the story owner at least twice a day

No shirking duties as pair

Knowledge sharing helps create
generalizing specialists

Timeouts, dojos, whiteboard sessions




KEEP IT WORKING

Rolling ownership


Day 1

Owner
▼



1/2 day
Stand-up

Kris Pirate Rob




1/2 day
After lunch

Odette Kris


Day 2

Owner
▼



1/2 day
Stand-up

Kev Odette



1/2 day
After lunch

Gus Kev

2-day story and 5 people worked on it



**KEEP
IT
TOGETHER**



Continuous integration

Always stay on the trunk and check in at least every 2 hours

Single code base

Only running tested code

Always run the build locally first



Automated build

Follow the money and 'ship' every iteration to production

Get value from tests

Virtualization and parallelization (test constraints)

Drive optimization and non-functional tests

RPMs



Build, front and centre

Keep the build status very visible using really big monitors around the bullpen

Pass / Fail - Fun with avatars

Selenium runs - Multiple browsers in parallel

Build stats / Hudson schedule

Periodic screen clear



KEEP IT TOGETHER



Build discipline

Make broken builds noisy and annoying

‘Stop the line’

Wear hats to indicate ownership

False negatives still require resolution



unit-test

BUILD 815



max steve dan

add unrealised pnl to pnl report



**KEEP
IT
REAL**



Continuous 'environmenting'

Team administers and supports all its environments

Stops team shipping crap to production

Team develops expertise in how product runs

Freedom to innovate in the 'system space' to meet business needs

Monitoring and alerting with Nagios and Munin



KEEP IT REAL

The screenshot displays the Nagios web interface with a sidebar on the left and a main content area on the right. The sidebar includes sections for Home, Monitoring, Reporting, and Configuration. The Monitoring section is active, showing a search for 'hostname' and a list of navigation options. The main content area features a 'Tactical Monitoring Overview' box with system information, a 'Network Outages' box showing 0 outages, and three summary tables: 'Hosts' (0 Down, 0 Unreachable, 26 Up, 0 Pending), 'Services' (0 Critical, 0 Warning, 0 Unknown, 190 Ok, 0 Pending), and 'Monitoring Features'.

Tactical Monitoring Overview
 Last Updated: Mon Mar 9 13:56:29 GMT 2009
 Updated every 20 seconds
 Nagios® 3.1.0 - www.nagios.org
 Logged in as ?

Network Outages
 0 Outages

Hosts

0 Down	0 Unreachable	26 Up	0 Pending
--------	---------------	-------	-----------

Services

0 Critical	0 Warning	0 Unknown	190 Ok	0 Pending
------------	-----------	-----------	--------	-----------

Monitoring Features

	Flap Detection	Notifications	Event Handlers	Active Checks	Passive Checks
Enabled	All Services Enabled	All Services Enabled	All Services Enabled	All Services Enabled	All Services Enabled
	No Services Flapping	All Hosts Enabled	All Hosts Enabled	All Hosts Enabled	All Hosts Enabled
	All Hosts Enabled				
	No Hosts Flapping				



KEEP IT REAL

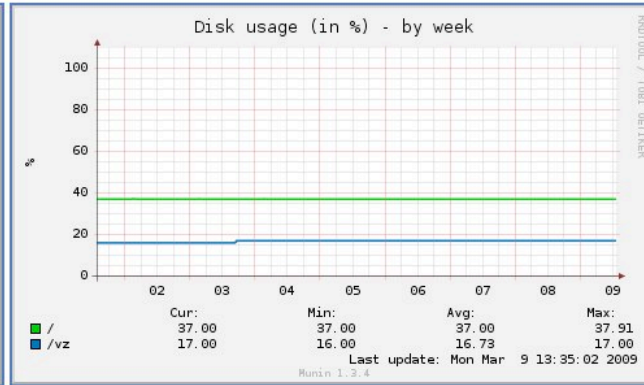
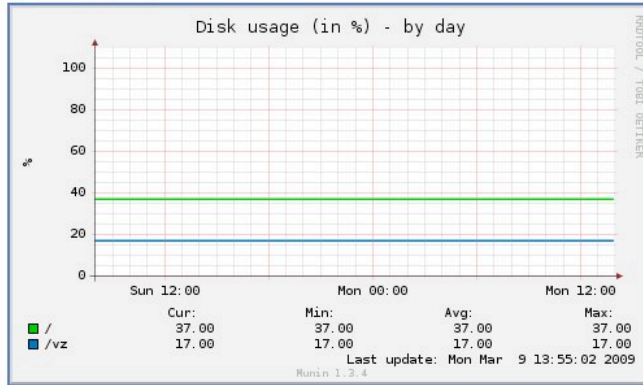


Overview :: [HardwareNodes](#) :: [hn4.energylab.ew](#)

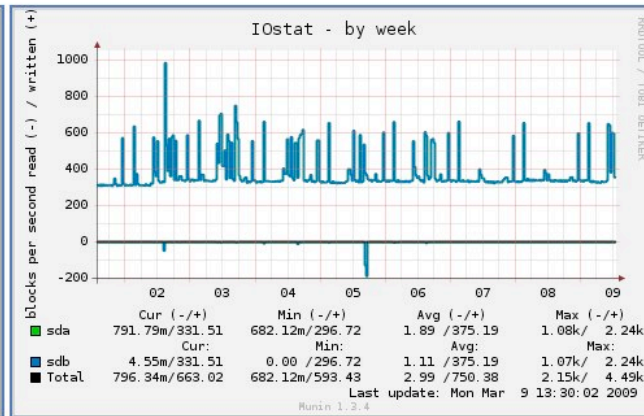
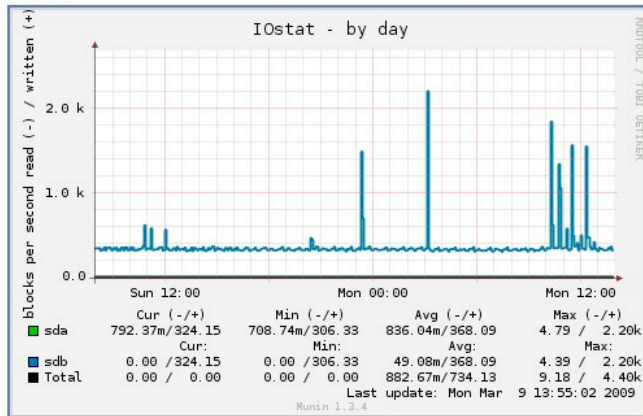
hn4.energylab.ew :: [[disk](#) [network](#) [other processes](#) [sensors](#) [system](#)]

disk

:: [Disk usage \(in %\)](#)



:: [IOstat](#)



Optimize at the right time

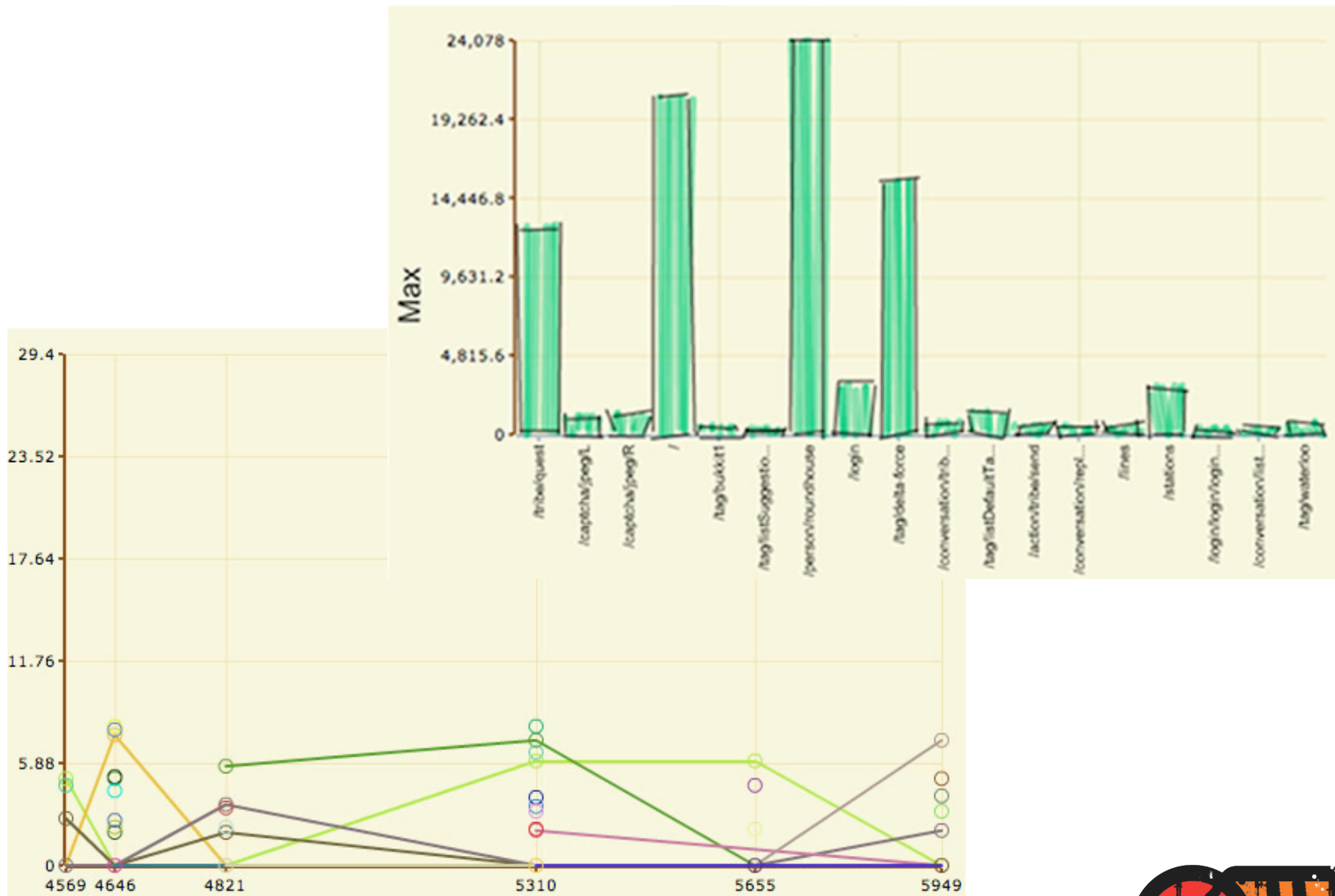
Make optimizations in the context of ‘the whole’

Play scenarios through the real system using Selenium, JUnit, JUnit, HttpUnit tests measuring with Jamm

View results in interactive graphs (Open Flash Chart 2, Flot, Google Charts) and compare before and after



KEEP IT REAL



```

26
27 void testPersonFailsLogin() {
28     WebResponse response = wc.getCheckedResponse('/login'.absolute(), 200)
29     WebForm loginForm = response.getFormWithID('loginForm')
30     loginForm.setParameter('username', username)
31     loginForm.setParameter('password', 'something wrong')
32     loginForm.setParameter('captchaText', 'xxx')
33     response = loginForm.submit()
34     assertEquals(1, response.getElementsWithAttribute('class', 'errors').length)
35     assertTrue(response.text.contains('Invalid username or password, or registration not confirmed'))
36 }
37

```

```

package com.energizedwork.grails.httpunit.scenario

import com.energizedwork.grails.httpunit.mock.*

scenarios = {
    SampleScenario([haltOnFailure:false]) {
        "${PassTest.class.name}" {
            testPass([sleep:100])
        }
        "${FailTest.class.name}" {
            testFail()
        }
        "${MultiPassTest.class.name}" {
            '*'([threads:10])
        }
    }
}

```



KEEP IT REAL

Fail over with confidence

Test failover regularly with automated tests

Tests monitoring and alerting

Tests redundancy at every level



That 'other stuff'

The operational stuff - all the non-code computer stuff - is a team responsibility

Sysadmin pairs with developers

Yellow cards

Test-driven

Manage external dependencies as partnerships



**KEEP
IT
COMING**



KEEP IT COMING

Charter a course

Set out a charter that visualizes where you're trying to get to

Vision and goals

'Futurespective' success criteria

High-level roadmap and Activity stories



DEC-2008 (BASIC TRADING PLAN)

1. Market

2. Entry

3. Exit

4. Risk Management

5. Psychology

6. Review

7. Journaling

8. Backtesting

9. Demo

10. Live

11. Mentorship

12. Community

JAN-2009 (MARGIN & PARTNER RESAYS)

1. Margin

2. Partner

3. Resays

4. Review

5. Journaling

6. Backtesting

FEB-2009 (ENRICHMENT)

1. Enrichment

2. Review

3. Journaling

4. Demo

5. Live

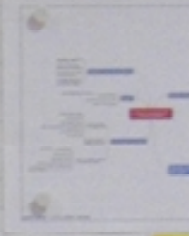
MAR-2009

1. Review

2. Enrichment

3. Journaling

4. Demo



5. Review

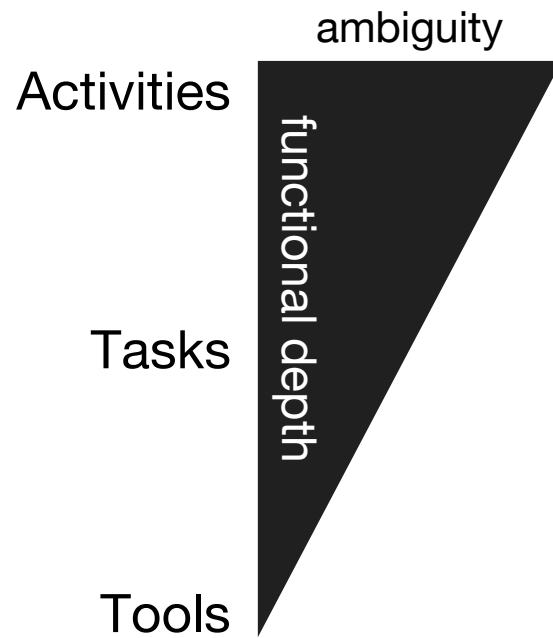
6. Journaling

7. Demo

8. Live

KEEP IT COMING

User stories



Cards



Jeff Patton, <http://agileproductdesign.com>





DISPLAY A LEDGER OF
THE TRANSACTION ON AN
INSTRUMENT.

OR
SC
DN
OR
PB
SC
DN

	Thu	Fri	Mon	Tue	
TODD	3	2	1	0	3

FOR CUSTOMER + HEDGE

3000 ON P/L CLOSE BLITTER IS UPDATED

✓ ✓ Δ POSITION = OLD POSITION - NET CHANGE DAILY OF P/CLOSE

✓ ✓ Δ REALISED P/L BASED ON P/CLOSE IS ADDED TO
CUMULATIVE P/L AND ALSO TO NET REALISED P/L

✓ ✓ Δ REALISED P/L TAKES INTO ACCT WHETHER POSITION WAS OPEN
BEFORE OR AFTER TRADE

✓ ✓ Δ UNREAL P/L IS BASED ON THE POSITION'S OPEN POSITION SIZE

✓ ✓ Δ CHECK VAC OPEN POSITION IS BASED ON REMAINING
OPEN POSITIONS

✓ ✓ Δ CHECK DAILY P/L CAPTURES THE TRADE, COMMISSION
& ROLLING P/L P/CLOSE TRADE

Give the backlog some luvin'

Onsite customer constantly 'works the board' to maximize profit

Watch and react to buffer levels

Just enough for 4 weeks (t-shirts)

Pairs with tester to evolve story cards from 'activities' to 'tasks' to 'tools'



MAR - 2009

S=1/2 M=1-2
L=2-4 XL=4-8

FEB - 2009
24 1/2 - 49



ITERATION # 11

START: 18/02/09
END: 24/02/09

VELOCITY: 7 1/2

4	3	2 Finished	1 cleaned up	4	3	2	NOT STARTED	IN PROGRESS	DONE
Assign customer of week? be notified that customer left to support other projects	Assign customer of week? be notified that customer left to support other projects	Read articles on new and test added to backlog	Assign customer of week	Run tests and fix bugs and update user stories	Assign customer of week and test added to backlog	Assign customer of week and test added to backlog	Assign customer of week and test added to backlog	Assign customer of week and test added to backlog	Assign customer of week and test added to backlog
Implement database schema Support schema for customer data, etc. Support schema for customer data, etc.	Implement database schema Support schema for customer data, etc. Support schema for customer data, etc.	Copy schema to test env (backup)	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env
		Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env
		Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env
		Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env
		DIVS	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env
		Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env
		Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env
		Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env
		Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env
		Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env	Test schema on test env

COP

BCIF

(NEED
SB, C
on 20

MANCO
OVER A

URGENT

URGENT

URGENT

URGENT

URGENT

URGENT

URGENT

Planning

Pull in more cards with an 'on-demand' planning game

Pre-planning - Onsite customer, tester and 2 developers review acceptance criteria / wireframes and sizes stories to be 1/2 - 3 days

'Pomodoro' planning - Whole team estimates stories in real time using planning poker



KEEP IT COMING

Showcase

Stimulate the client to decide whether to fund more iterations, or not

Same time, same place - Tuesday at 16:30

Sponsor and onsite customer always attend

Protect the showcase by managing the run-in

PR Event - Rehearse. Compered by onsite customer.
Last owner demos story.





**WE MAXIMIZE PROFIT BY
'SHIPPING' RUNNING TESTED
FEATURES TO PRODUCTION AT
LEAST EVERY WEEK**



**AUTOMATION GIVES US
MOBILITY SO WE
CONTINUOUSLY INVEST IN IT
TO KEEP IT CHEAP**



**WE DON'T TRACK BUGS
WE FIX 'EM**



WE MANAGE DEBT SO THAT WE CAN KEEP MOVING FAST



**WE'RE COLOCATED BECAUSE
THE CONVERSATIONS NEVER
STOP**





**WE LEARN WHAT'S NEEDED
BY ITERATING WITH A
FULL-TIME ONSITE CUSTOMER
AT THE WHEEL**



**OUR PEOPLE ARE 'CHILLED'
AND OUR TEAM IS RESILIENT
BECAUSE WE PAIR-PROGRAM
ALL THE TIME**



**WE MAINTAIN A RIGOUROUS
TEST-DRIVEN APPROACH
THROUGH OUR PERSONAL
DISCIPLINE**



THE BUILD MONITORS ARE OUR FOCAL POINT



**IF WE BREAK THE BUILD WE
FIX IT IMMEDIATELY,
OTHERWISE WHAT'S THE
POINT OF HAVING IT**



WE OPERATE ALL OUR ENVIRONMENTS, INCLUDING PRODUCTION



**NON-FUNC AND SYSADMIN
WORK IS PART OF OUR
PRODUCT DEVELOPMENT AND
NOT AN AFTERTHOUGHT**



**WE WRITE STORIES TO BE A
LITTLE AMBIGUOUS TO
ENSURE CONVERSATION
HAPPENS**



**WE PLAN JUST ENOUGH
WHEN WE NEED IT TO
PREVENT US STALLING**



**WE MAKE THE SHOWCASE
COUNT BECAUSE OUR CLIENT
WILL DECIDE WHETHER TO
INVEST MORE MONEY**



**HIRE THE RIGHT PEOPLE
AND THEY'LL CREATE THE
RIGHT ENVIRONMENT**



WORK WITH THE RIGHT CLIENTS



KEEP IT MOVING

KEEP IT WORKING

KEEP IT TOGETHER

KEEP IT REAL

KEEP IT COMING

